# SURVEY - Multicriteria Models for Just-in-Time Scheduling


| | |
|---|---|
| Journal: | *International Journal of Production Research* |
| Manuscript ID: | TPRS-2009-IJPR-0513.R1 |
| Manuscript Type: | State-of-the-Art Review |
| Date Submitted by the Author: | 22-Feb-2010 |
| Complete List of Authors: | T'Kindt, Vincent; Universite' Francois-Rabelais, E3i |
| Keywords: | MODELLING, JIT, MULTI-CRITERIA DECISION MAKING |
| Keywords (user): | JIT Scheduling, Multicriteria scheduling |

*International Journal of Production Research*
Vol. 00, No. 00, 15 March 2008, 1–18

## SURVEY

# *Multicriteria Models for Just-in-Time Scheduling*

T'kindt Vincent

*Université François-Rabelais de Tours, Laboratoire d'Informatique, 64 avenue Jean Portalis, 37200 Tours, France.*

(*revision 1, January 2010*)

Just-in-Time manufacturing consists in organizing the production of elements in order to meet a certain number of objectives or requirements according to the so-called "Just-in-Time philosophy". Just-in-Time has been extensively studied in the literature for many years due to the high number of real-life situations where it can be applied. This paper aims at revisiting Just-in-Time principles and detailing how they can be applied to the scheduling stage of a manufacturing process. Therefore, new models which are multicriteria ones by their very nature, are presented and discussed. The conclusions highlight the fact that most of the existing models presented in the scheduling literature happen to be incomplete regarding Just-in-Time principles.

**Keywords:** Just-in-Time scheduling; Multicriteria scheduling; Models

## 1. Introduction

**Production management has been a research subject for many researchers and engineers over the years and several general policies, philosophical principles, have been set up. Just-in-Time manufacturing (JiT) is one of those resulting principles. JiT manufacturing concentrates on the production stage and advocates the elimination of wastes by optimizing the manufacturing process. This includes optimizing the organization of the shop, the relations with customers and the production process. As outlined by Kannan and Tan (2005), JiT principles improve the production process of a company under the condition of a good integration of customers and suppliers into the production process. From an historical point of view, interest in JiT manufacturing appeared after the Second World War in the Toyota factories (see a short description in Pinedo and Chao (1999)) which had to meet due dates for orders whilst dealing with the non negligible storage cost of the orders produced.**

**This paper deals with production scheduling and concentrates on JiT principles. Scheduling consists in deciding on the allocation of tasks on resources over time in order to optimize a given number of objectives also called criteria (T'kindt and Billaut 2006). A cursory glance at the literature shows that two types of papers dealing with JiT scheduling**

*Corresponding author. Email: tkindt@univ-tours.fr

ISSN: 1478-6451 print/ISSN 1478-646X online

DOI: 10.1080/1478645YYxxxxxxx
http://www.informaworld.com

**exist. There are papers mainly focusing on JiT production and papers concentrating on solution techniques applied to JiT scheduling. Generally the former take account of JiT principles and are often related to real-life applications. The latter, on the contrary, focus on more or less simple models for which very dedicated scheduling algorithms are provided. However, these models can be far from the reality of JiT production. The aim of this paper is, first, to propose a general formulation, a multicriteria one by its very nature, that answers all the requirements of a JiT production system (section 2). Next, literature on JiT scheduling is revisited in the light of the previous formulation (section 3) and new models are proposed (section 4).**

Notice that this paper is not dedicated to solution algorithms but focuses more on models, existing or new ones, and their links and impacts in terms of a JiT policy. The main contribution also results from the gathering of several features related to lot-streaming and multicriteria optimization which has never been studied before.

## 2. Multicriteria Just-in-Time scheduling

### 2.1 *General Just-in-Time principles*

Numerous definitions of a JiT production system can be found in the literature. Among others, Nollet *et al.* (1994) describe such a system as one which "processes and delivers finish goods *just-in-time* to be sold, components *just-in-time* to be assembled into finished goods and materials bought *just-in-time* to be converted into components". In a JiT production system, quality and productivity have to be improved at all stages of the industrial system. This implies reducing wastes and taking into account human factors (Nollet *et al.* 1994). This reduction is the main challenge when implementing a JiT system and the term "waste" has usually to be defined depending on the real production system. However, Nollet *et al.* gather a series of general elements as potential sources of wastes.

There are *wastes due to overproduction* which induce wasteful storage costs, increased human requirements, *etc.*, opposed to *wastes due to waitings* caused by machine breakdowns for instance. There are *wastes due to an inadequately organized shop floor* (wasteful transportation and material handling, for instance, when two resources are too far from each other) or *wastes due to a failing or badly prepared production process*. In the same vein, there can be *wastes due to production flaws*. At last, and this is of a high importance at the planning and scheduling phase, there are *wastes due to the storage of in-process or finished goods*. This is a crucial point in a JiT policy.

Another way to define briefly the JiT philosophy, which is complementary to that of Nollet *et al.*, is given by Baglin *et al.* (2001): each product must ideally be processed on a "chain of machines". This means that when an order (a job) enters the shop it has to be processed by the machines ideally without waiting time, as if they were available for it alone. This is the *smoothing* of the job flow. Clearly, each machine must also have a smooth flow of jobs to process in order to be made cost-effective which, in a sense, can be conflicting with the smoothing of job flow.

All the above elements are production based, but in this paper we focus on the scheduling component of the production system. Therefore, only a subset of the above quoted notions concerns scheduling. Firstly, the notion of "chain of machines" can be easily translated into the "no-wait" constraint of the classic scheduling theory (see T'kindt and Billaut (2006) for a formal definition of that constraint). However, in the case of "pretty close" due dates, imposing the no-wait constraint may result in increasing the tardiness of products, and therefore customer dissatisfaction. Consequently, we may have to violate this constraint and thus increase in-process storage costs in order to limit the tardiness in producing orders. This means that strictly processing products on a "chain of machines" is a concept that may conflict with that of limiting wastes due to the storage of raw materials and customer dissatisfaction. Limiting wastes caused by the storage of materials is, as quoted by Nollet *et al.* (1994), a key point in a JiT production system. Storage is related to three distinct elements of production: the raw materials, the in-process and subcontracted components, and the finished products. The aim is, therefore, to improve the quality and productivity at each level where these elements are encountered in order to reduce the induced storage costs and to answer as much as possible the lead times (Schonberger 1982).

Now let us consider the JiT principles. First of all, one should notice that it is merely impossible to propose a very general JiT model: we just aim at proposing a quite general JiT formulation, limited to the scheduling component, in the light of general JiT principles.
Firstly, let us consider the arrival of raw materials on the shop floor. Due to contractual aspects and to the production for suppliers of raw materials, the need for these ones is often evaluated at a mid-term planning level, *i.e.* long before the scheduling phase. Based on the routings and the decomposition of a product, the required raw materials and components are often ordered independently of the scheduling phase. Hence, during the Material Requirements Planning phase in a MRP system, we decide which materials will be made available in the shop, in which quantity and when. As in this phase we do not have an accurate view of what will be the real operations schedule, so materials are usually made available in the shop before the start of such a schedule (or sufficiently early before an operation, requiring materials, starts). Thus, the calculation of a schedule in a JiT environment can be achieved by considering that each job has a release date corresponding, at least, to the arrival of the raw materials or components in an assembly system. Consequently, this one can correspond to the maximum date among the delivery times negotiated with the suppliers of the required raw materials. Release dates can also occur for internal reasons, as for instance when the production has been balanced in time at the medium-term planning phase. In this case, release dates can be dispatched to take into account the limited production capacities.
The situation for in-process and subcontracted components is different, notably for work-in-process components because they induce storage constraints and costs which are directly related to the operations schedule. This is also the case for finished products. It follows that, when calculating a JiT schedule, the limitation of work-in-process and finished products storage must be taken into account. As a consequence, reducing storage leads to reduce lead times and products tend to be produced on a "chain of machines". At the end of the chain, the customer is taken into account via the customer dissatisfaction, *i.e.* a measure of the potential cost implied by delivering a job late, regarding the contractual due date of the job.

**Regarding the limitation of work-in-process and the reduction of lead times it can be interesting, whenever possible, to consider at the scheduling phase the job lot-streaming. Lot-streaming consists in splitting each operation of a job into sublots with the condition that a sublot can be**

**started on the next machine in the routing without waiting for the completion of the whole operation on the current machine. Lot-streaming combines lot splitting and operations overlapping, and is induced by the capability of a production system to transfer part of an operation. Clearly, lot-streaming is a matter of scheduling and can result in reducing the time needed to produce a job and in reducing the time spent by an operation in a storage area. It follows that we might be interested in considering the lot-streaming constraint in the context of a JiT scheduling model.**
In the remainder of this section we first review literature on lot-streaming scheduling models before developping a mathematical formulation of the costs to be minimized when calculating a JiT schedule.

### 2.2 *The lot-streaming constraint in scheduling*

It is interesting to notice that in the case of *divisible* jobs (*i.e.* jobs made up of divisible parts) the lead times can be reduced by enabling lot-streaming. The latter consists in transferring any sublot of an operation (*i.e.* a part of the job) to the next machine without waiting for the completion of the whole operation (see Figure 1). All sublots of the same job must be sequentially processed on any given machine, and between two of these sublots voluntary idle times can be inserted if this helps reducing the costs. The problem of determining the number and size of sublots per job is called the *lot-streaming problem.*

Insert Figure 1 about here

**Lot-streaming has been the subject to various studies in the scheduling field along the years. A comprehensive survey of lot-streaming based problems and algorithms can be found in Baker and Jia (1993), Potts and van Wassenhove (1994), Triesch and Baker (1994), Chang and Chiu (2005) and Sarin and Jaiprakash (2007). These surveys show the diversity in the use of the lot-streaming constraint and that a classification of problems, encountered in practice or in the literature, can be established depending on several factors. The first one is the sublot configuration, which enables us to distinguish between: problems with different number and size of sublots per job, problems with** *consistent* **sublots for which the number of sublots is equal for all operations, problems with equal-size sublots for which all operations of the same job are decomposed into equal-size sublots, etc. Scheduling problems with lot-streaming can also be distinguished according to the presence or not of setup times before each sublot, of transfer times between sublots on two consecutive processing machines or of possible inserted idle times between sublots. The literature on lot-streaming and scheduling is rich of publications, but it appears that most of them deal with the minimization of the makespan. To the best of our knowledge, the only work dealing with the lot-streaming constraint in JiT scheduling is due to Yoon and Ventura (2002b) who consider a flowshop scheduling problem with earliness and tardiness penalties. They notably show that, when the sequence of jobs is given, the lot-streaming problem can be solved in polynomial time by means of linear programming for various sublot configurations (equal-size sublots and consistent sublots) and various additional constraints (no-wait constraint, blocking constraint,**

**limited capacity buffers, ...).**

On the solution side, the sequencing and lot-streaming problems are rarely solved together. Generally, due to the complexity of these problems, they are solved separately in a heuristic iterative scheme. An example of solution of this kind of problems can be found in Dauzère-Pérès and Lasserre (1997) which deals with the jobshop scheduling problem with makespan minimization. The authors consider the case of consistent sublots, *i.e.* all the jobs are split into the same number of sublots, but for an operation all the sublots may have a different size. The main line of their heuristic algorithm is as follows. First, lot sizes are fixed and the sequencing problem is solved by means of an existing heuristic for the jobshop problem. Next, the sequence is assumed to be fixed and the lot-streaming problem is solved via the solution of a linear program. Starting for the obtained lot sizes, the process sequencing/lot-streaming is iterated until no improvement on the makespan criterion is obtained. The computational experiments proposed in Dauzère-Pérès and Lasserre (1997) show that this heuristic provides good results.

**In this paper, we consider the case of equal-size sublots with no transfer time, no setup time between two sublots of the same operation, and possible inserted idle times. Such assumptions have been considered in most of the scheduling problems with lot-streaming dealt with in the literature (Sarin and Jaiprakash 2007). Besides, and as argued by Ramasesh *et al.* (2000), "the assumption of equal-size sublots is made in the interest of analytical tractability and is clearly justifiable and practical". Consequently, the model proposed in the remainder can be easily generalized to other sublot configurations as for instance consistent sublots.**

### 2.3 *Modelling of production costs in JiT scheduling for shop problems*

Let us define the general scheduling problem under consideration. Assuming that $n$ jobs $J_i$ have to be scheduled on a set of $m$ machines, the routing for job $J_i$ is defined by $\pi_i = (\pi_i(1); \pi_i(2); ...; \pi_i(m))$ with $\pi_i(k)$ the number of the $k$-th machine which processes $J_i$. Whenever sequences $\pi_i$, $\forall i = 1, ..., n$, are fixed, we face a jobshop or flowshop problem and if determining these sequences is a part of the problem, then we face an openshop problem.
**Besides, we assume that each job $J_i$ is made up of $q_i$ indivisible elements and can be decomposed into equal-size sublots according to the lot-streaming constraint. This implies that all jobs are split into $\delta$ sublots with $\delta$ a variable to compute and the $q_i$'s being part of the data.**
This assumption enables us to be more general than in classic scheduling where jobs are often *indivisible*, *i.e.* $q_i = 1$, $\forall i = 1, ..., n$. We define jobs as *divisible*, if $q_i > 1$, $\forall i = 1, ..., n$ and in that case a fixed division cost, referred to as $\lambda_i$, exists for each job $J_i$. The processing of the $q_i$ elements of job $J_i$ on machine $M_j$ requires a processing time equal to $p_{i,j}$ and we assume that the processing of one element is equal to $\frac{p_{i,j}}{q_i}$. The completion time of job $J_i$ on machine $M_j$ is denoted by $C_{i,j}$ and is a result of the schedule.
As outlined in section 2.1 each job $J_i$ has a release date, denoted by $r_i$, which corresponds to the arrival date of the raw materials requested for the processing of that job. Similarly, a due date $d_i$ is set up and corresponds to the delivery date to the customer. We denote by $\beta_i$ the unit cost for completing job $J_i$ tardy, *i.e.* whenever $J_i$ is delivered at a completion time $C_{i,\pi_i(m)} > d_i$ a cost equal

to $\beta_i T_i = \beta_i \max(0; C_{i,\pi_i(m)} - d_i)$ is generated. This cost corresponds to the customer dissatisfaction. Similarly, the earliness, denoted by $E_i$, is defined by $E_i = \max(0; d_i - C_{i,\pi_i(m)})$ and occurs if $C_{i,\pi_i(m)} < d_i$.
At last, we refer to $\gamma_i^{\pi_i(j)}$ as the unit storage cost of work-in-processes of job $J_i$ between machines $M_{\pi_i(j)}$ and $M_{\pi_i(j+1)}$ and to $\kappa_i$ as the unit storage cost of finish product of job $J_i$.

Insert Table 1 about here

A summary of the previously presented data is provided in Table 1. We now focus on defining the cost associated to the processing of job $J_i$ and we define $t_{i,j,k}$ as the starting time of the $k$-th sublot of job $J_i$ on machine $M_{\pi_i(j)}$.
The total cost, referred to as $Z_i$, induced by a job $J_i$ is made up of the cost for the storage of work in-processes (Figure 2), the storage of finished products (Figure 3), the cost for splitting into sublots and the cost for delivering $J_i$ tardy. This cost was first established in T'kindt and Billaut (2006).

Insert Figure 2 about here

Insert Figure 3 about here

**Lemma 2.1:** *The total scheduling cost of job $J_i$ when produced Just-in-Time is defined by*

$$\begin{aligned} Z_i = \; & \beta_i T_i + \delta\lambda_i + \kappa_i q_i E_i + \kappa_i q_i \frac{p_{i,\pi_i(m)}}{\delta} + \kappa_i q_i t_{i,\pi_i(m),\delta} \\ & - \frac{q_i}{\delta} \sum_{j=1}^{m} (\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}) \sum_{k=1}^{\delta} t_{i,\pi_i(j),k} \\ & - \frac{q_i}{2\delta} \sum_{j=1}^{m} (\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}) p_{i,\pi_i(j)}. \end{aligned}$$

First of all, notice that this cost is not a linear function in the general case, since like the starting times $t_{i,j,k}$, the number of sublots $\delta$ has also to be computed.
The first three terms of this cost are the weighted tardiness, the running cost and the weighted earliness, respectively. The fourth and last terms are only functions of the number of sublots $\delta$ whilst the fifth and sixth both depends on the number of sublots and on the starting times. If the number of sublots is fixed, then the cost turns to a polynomial function of the starting times (the earliness and tardiness can also be expressed as a function of the starting time of the last sublot in the last machine of the routing of each job).

We are now ready to state the general multicriteria JiT scheduling problem which we refer to as $(P_{JiT})$.

$$\begin{aligned} & \text{Minimize } Z_1(s) \\ & \text{Minimize } Z_2(s) \\ & \ldots \\ & \text{Minimize } Z_n(s) \\ & \text{sc} \\ & \quad t_{i,\pi_i(1),1}(s) \geq r_i, \; \forall i = 1, \ldots, n \\ & \quad s \in S \end{aligned}$$

with $S$ the set of solutions defined by the constraints of the scheduling problem. From lemma 2.1 it is clear that the costs $Z_i$ and $Z_j$ for two jobs are likely to be conflicting as soon as their due dates are sufficiently close, since in that case, in order to minimize the cost of one of these jobs, we may have to delay the other

one and induce storage costs or dissatisfaction (tardiness) costs. Consequently, for a decision maker the aim seems to find a good compromise between all the costs $Z_i$ in order to be *as much as possible Just-in-Time for all jobs.* This latter notion clearly depends on the preference on the jobs of the decision maker. It is a real matter of decision aid.
In this paper we consider that finding a good compromise solution for all the costs is equivalent to finding a strict Pareto optimum for the $Z_i$'s. We say that a schedule $s$ is a strict Pareto optimum, if there does not exist another schedule $s'$ such that $Z_i(s') \leq Z_i(s)$, $\forall i = 1, ..., n$, with at least one strict inequality.
On a practical side, minimizing simultaneously all the $Z_i$'s doesn't make much sense due to the high number of criteria. Henceforth, model $(P_{JiT})$ only serves to analyse theoretical aspects of practical JiT problems. For instance, it may help in deciding wether it is relevant or not to minimize the sum of the earliness and tardiness costs for various machine configurations (see section 3). Here, *relevant* means that we fulfill the requirement of a JiT prodution system and that the optimal solution of the aggregated problem is not dominated, regarding the criteria $Z_i$, by another solution.

In the remainder we investigate the existing JiT models in the scheduling literature and we propose links with the above costs $Z_i$ and the calculation of strict Pareto optima.

## 3. Review of Just-in-Time scheduling literature

Just-in-Time scheduling literature has been the subject of numerous state-of-the-art reviews (Baker and Scudder 1990, Hall and Posner 1991, Gordon *et al.* 2002a,b, Kaminsky and Hochbaum 2004, Gordon *et al.* 2004). These reviews show that many models for JiT scheduling have been considered over the years and that often they have been based on the costs for completing jobs early or tardy. These models have been introduced without any practical justification nor any link with the JiT production philosophy. This lack of connection is certainly the main reason for the diversity of JiT scheduling models studied in the literature. Few works try to integrate some JiT features other than the customer dissatisfaction or the storage cost for finished products. Bulbul *et al.* (2004) consider the case of indivisible jobs (*i.e.* without the lot-streaming constraint) in a flowshop problem in which the sum of the earliness cost, the tardiness cost and the intermediate inventory holding cost are minimized. The minimized objective function is equal to the sum of criteria $Z_i$ defined in section 3.1 for a flowshop problem and therefore their problem can be considered as a special instanciation of the general multicriteria model described in this paper. The authors propose heuristics and a lower bound based on mathematical programming. Shi and Pan (2003) consider a slightly different jobshop JiT scheduling problem for which the aim is to minimize the total inventory holding cost, in the case of indivisible jobs, as included in the definition of the $Z_i$'s in section 3.1. To guarantee the delivery dates, they do not minimize the weighted tardiness criterion but consider that each job has a deadline. They propose heuristic algorithms to solve the problem.

One of the aim of this paper is to review classic JiT scheduling models in the light of the general multicriteria model deduced from the literature on JiT production systems (see section 2.3).
Nevertheless, it is interesting to notice from previous literature reviews, that JiT scheduling problems are complex scheduling problems which basically involve

three subproblems: a *due date quotation problem*, an *optimal timing problem* and a *sequencing and assignment problem*. All these problems should be considered simultaneously when solving a JiT scheduling problem. However, for practical reasons they are often solved separately in order, for instance, to obtain a heuristic solution. In the remainder of this section we briefly focus on the two particular problems which are the due date quotation problem and the optimal timing problem. This short review aims at highlighting some sub-problems to deal with when solving JiT scheduling problems even for the newly introduced models.

The *due date quotation problem* is a highly strategic problem since it is related to the customer. Therefore, fixing due dates may not be just a matter of science, but also commercial one. For some of the problems found in the literature, the due dates are assumed to be fixed and the due date quotation problem has been solved before the scheduling phase. For other problems, this is not the case and several classic rules exist to solve the due date quotation model as for instance the CON, SLK, TWK, NOP and PPW rules (see Kaminsky and Hochbaum (2004) for a recent review). Sometimes, no rule is used and the due dates are just variables of the scheduling problem. There are also numerous problems for which all jobs are assumed to share the same common due date.
This range of situations illustrates well how many different situations can occur in real-life applications and how the due date quotation problem is of a high importance.

The *optimal timing problem* is related to the solution of the subproblem for which we assume that the sequences of jobs on machines are known. Consequently, the optimal timing problem consists in calculating the optimal starting times $t_{i,j,k}$ and has an interest since the objective function is often not a *regular function*[1]. This problem can most of the time be solved in polynomial time even if the original JiT problem is NP-hard.
Historically, the first paper dealing with an optimal timing problem is due to Garey *et al.* (1988) who solved a very particular single machine problem with only earliness and tardiness penalties. This algorithm has inspired several authors who next proposed extensions to it. The first ones were Szwarc and Mukhopadhyay (1995) who considered an extension to the weighted earliness and tardiness case. For parallel machines with precedence constraints, Della Croce and Trubian (2005) proposed a polynomial time algorithm for the earliness and tardiness costs.
More recent works exploit properties of convexity or even linearity of the cost functions, which yields to more general or faster algorithms. This is the case of Pan and Shi (2005) for the problem tackled by Garey et al. who propose a faster solution procedure. Chrétienne and Sourd (2003) consider the general problem with operations connected by precedence constraints and such that each operation has a convex cost function. The aim is to minimize the sum of the convex cost functions. This problem has numerous applications including JiT problems with earliness and tardiness penalties. In the particular case of a single machine with convex cost functions, Hendel and Sourd (2007) refine the algorithm of Chrétienne and Sourd (2003) and propose an extension to the flowshop problem where there is only a cost function on the last operation of each job. Sourd (2005) considers the single machine problem for which the cost functions of each job are assumed to be piecewise linear. The aim is to minimize the sum of the cost functions together with the cost of idle times. This latter cost is defined as the sum of the costs

[1] An increasing function of the completion times.

induced by letting the machine idle between two consecutive jobs. Therefore, the two costs are likely to be conflicting. This problem is $\mathcal{NP}$-hard in the weak sense and a dynamic programming algorithm is proposed.

We can now turn to a literature review, which separates problems with indivisible jobs from problems with divisible jobs. To refer shortly to JiT scheduling problems, we use the three-field notation introduced and extended by T'kindt and Billaut (2006) to multicriteria scheduling problems by Graham *et al.* (1979).

### 3.1 *Problems with indivisible jobs*

Assume that all jobs are indivisible, *i.e.* $q_i = 1, \forall i = 1, ..., n$ and therefore $\delta = 1$. This is the case, mainly studied in the literature, for which no lot-streaming can be implemented. The total cost of job $J_i$ given in lemma 2.1 is particularized in the following corollary.

**Corollary 3.1:** *In the case of indivisible jobs, i.e.* $q_i = 1, \forall i = 1, ..., n$, *the total scheduling cost of job* $J_i$ *when produced Just-in-Time is defined as*
$Z_i = \quad \beta_i T_i + \kappa_i E_i + \lambda_i + \kappa_i C_{i,\pi_i(m)} - \sum_{j=1}^{m} (\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}) C_{i,\pi_i(j)}$.

**Proof:** (sketch). Apply lemma 2.1 with $q_i = 1, \gamma_i^{\pi_i(0)} = 0, \gamma_i^{\pi_i(m)} = \kappa_i, \forall i = 1, ..., n$.
□

For optimisation purposes, the term $\lambda_i$ can be omitted from the previous formula since it is a constant, therefore leading to the following criteria to minimize
$Z_i = \quad \beta_i T_i + \kappa_i E_i + \kappa_i C_{i,\pi_i(m)} - \sum_{j=1}^{m} (\gamma_i^{\pi_i(j)} - \gamma_i^{\pi_i(j-1)}) C_{i,\pi_i(j)}$.
In the remainder of this section we consider this second formulation. Now, let us turn to the particularization of this cost to various machine environments. The following corollary presents two particular cases of corollary 3.1.

**Corollary 3.2:** *In the case of single-operation jobs (single machine and parallel machines environments), i.e.* $m = 1$, *the total job scheduling cost* $Z_i$ *is given by*

$$\beta_i T_i + \kappa_i E_i.$$

*In the case of 2-operation jobs (flowshop, jobshop and openshop environments), i.e.* $m = 2$, *the total job scheduling cost* $Z_i$ *is given by*

$$\beta_i T_i + \kappa_i E_i + \gamma_i \big(C_{i,\pi_i(2)} - C_{i,\pi_i(1)}\big)$$

**Proof:** (sketch). Apply corollary 3.1 with $m = 1$ for the single-operation jobs problem and $m = 2$ for the 2-operation jobs problems.
□

We now analyse the main objective functions studied in the JiT scheduling literature and known under the name of early/tardy models.
Early/tardy models all involve two criteria which can be: $\bar{E}$ and $\bar{T}$, $\kappa\bar{E}$ and $\beta\bar{T}$, or $\bar{E}^\kappa$ and $\bar{T}^\beta$. Generally, these criteria are gathered into a single linear objective function, referred to as $F_\ell(X, Y)$. Single machine and parallel machines environments have also been largely dealt with in the literature. Consider the following 2-job instance of the $1|d_i, \beta_i, \kappa_i|F_\ell(\bar{E}^\kappa, \bar{T}^\beta) = \sum_i \kappa_i E_i + \beta_i T_i$ problem.

| $i$ | $r_i$ | $p_i$ | $d_i$ | $\kappa_i$ | $\beta_i$ |
|---|---|---|---|---|---|
| 1 | 5 | 4 | 12 | 2 | 4 |
| 2 | 7 | 2 | 10 | 7 | 3 |

The set of strict Pareto optima for criteria $Z_1$ and $Z_2$ is depicted in Figure 4 together with the criteria vectors associated with the optimal solutions for the $F_\ell(\bar{E}^\kappa, \bar{T}^\beta) = \sum_i \kappa_i E_i + \beta_i T_i$ objective function.

Insert Figure 4 about here

The $F_\ell(\bar{E}^\kappa, \bar{T}^\beta) = \sum_i \kappa_i E_i + \beta_i T_i$ function represents here a total cost of producing Just-in-Time. Consequently the minimal total cost, which is equal to 8 in this example, is assumed to be representative of the best solution for the decision maker. However, he can refuse this solution for different reasons: job $J_1$ has a too high individual cost $Z_1$, or even this job implies a too long storage time, ... Therefore, consequences at job level may lead a decision maker to prefer a solution that does not minimise the total cost. This remark yields the conclusion that, sometimes, it can be interesting to analyse the JiT aspect of a schedule at the level of the $Z_i$'s and consequently to optimise those criteria instead of the weighted sum. We can definitely assume that optimising the $Z_i$'s offers more *flexibility* in the choice of the best solution by a decision maker. In other words, considering the $Z_i$'s and integrating the decision maker's preferences may lead to a different aggregated problem than that of minimizing the sum of the weighted earliness and the weighted tardiness.

What happens in the case of multi-operation jobs, for instance in the case of flowshop, jobshop or openshop problems? Let us consider the following 3-job instance of the $F2|prmu, d_i, \beta_i, \kappa_i|F_\ell(\bar{E}^\kappa, \bar{T}^\beta) = \sum_i \kappa_i E_i + \beta_i T_i$ sample problem.

| $i$ | $r_i$ | $p_{i,1}$ | $p_{i,2}$ | $d_i$ | $\kappa_i$ | $\beta_i$ |
|---|---|---|---|---|---|---|
| 1 | 5 | 4 | 4 | 16 | 2 | 4 |
| 2 | 7 | 2 | 3 | 15 | 7 | 4 |
| 3 | 6 | 4 | 5 | 17 | 5 | 5 |

The optimal schedule for the $F_\ell(\bar{E}^\kappa, \bar{T}^\beta) = \sum_i \kappa_i E_i + \beta_i T_i$ objective function, has a value of 30 and is depicted in Figure 5.

Insert Figure 5 about here

Schedule $\pi = (1, 2, 3)$ of Figure 5 is optimal for the provided instance of the 2-machine flowshop problem. Its objective function value is equal to 30 and is due to jobs $J_2$ and $J_3$ which are late and job $J_1$ which is early. But now, consider the JiT policy as defined in corollary 3.2 with in-process inventory costs $\gamma_i$'s defined as $\gamma = [1; 100; 1]$. We can see that job $J_2$ has a one unit waiting time between the two machines in $\pi$ and the $Z_i$'s are given by $Z_1(\pi) = 10$, $Z_2(\pi) = 404$ and $Z_3(\pi) = 25$. Now let us consider the schedule $\pi' = (1, 3, 2)$ given in Figure 6.

Insert Figure 6 about here

From the point of view of the $F_\ell(\bar{E}^\kappa, \bar{T}^\beta) = \sum_i \kappa_i E_i + \beta_i T_i$ objective function, schedule $\pi'$ is not optimal since the corresponding objective function value is equal to 35. But, the $Z_i's$ are given by $Z_1(\pi') = 10$, $Z_2(\pi') = 324$ and $Z_3(\pi') = 10$. In fact, $\pi'$ weakly dominates $\pi$, in the sense of Pareto optimality, regarding a real JiT policy defined by the $Z_i$'s. This is due to the fact that work-in-process costs (which must be reduced in a JiT production system) are not taken into account in the basic early/tardy model.
This simple example shows that in some cases the very well-known early/tardy model can lead to calculate *solutions which do not exactly follow a JiT policy.*

A straight extension of the early/tardy model consists in minimising criteria $\bar{E}^{\kappa}$, $\bar{T}^{\beta}$ and $\bar{C}^{\alpha}$, the latter being introduced as "a measure of the work-in-process", *i.e.* minimising the criterion $\bar{C}^{\alpha}$ reduces in a sense the work-in-process storage costs since it reduces the average time required by a job in the shop. However, this can only be a rough evaluation of these costs but not their exact measure. In that way, again, considering the criteria $Z_i$'s introduced in section 2.3 instead of these three criteria leads to a more accurate model of a JiT system (just consider again the previous 2-machine flowshop example).

Literature reviews about problems, results and algorithms to solve JiT scheduling problems with indivisible jobs can be found in Baker and Scudder (1990), Hall and Posner (1991), Gordon *et al.* (2002a,b), Kaminsky and Hochbaum (2004), Gordon *et al.* (2004).

### 3.2 *Problems with divisible jobs*

In this section we assume that all jobs are divisible into $\delta$ equal-size sublots and that the JiT job cost $Z_i$ is defined in lemma 2.1. To the best of our knowledge, this case has never been studied in the literature. We first present here particularizations of this lemma to some dedicated shop environments before providing a brief literature review.

**Corollary 3.3:** *In the case of single-operation jobs (single machine and parallel machines environments), i.e. $m = 1$, the total job scheduling cost $Z_i$ is given by*

$$\beta_i T_i + \kappa_i q_i E_i + \delta\lambda_i + \kappa_i q_i C_{i,1} - \frac{q_i \kappa_i}{\delta}\left(\sum_{k=1}^{\delta} t_{i,1,k} + \frac{p_{i,1}}{2}\right).$$

*In the case of 2-operation jobs (flowshop, jobshop and openshop environments), i.e. $m = 2$, the total job scheduling cost $Z_i$ is given by*

$$\beta_i T_i + \kappa_i q_i E_i + \kappa_i q_i C_{i,\pi_i(2)}$$
$$-\frac{q_i}{\delta}\left(\sum_{k=1}^{\delta}(\gamma_i t_{i,\pi_i(1),k} + (\kappa_i - \gamma_i) t_{i,\pi_i(2),k}) + \frac{\gamma_i p_{i,\pi_i(1)} + (\kappa_i - \gamma_i) p_{i,\pi_i(2)}}{2}\right)$$

**Proof:** (sketch). Apply lemma 2.1 with $\gamma_i^{\pi_i(0)} = 0, \gamma_i^{\pi_i(m)} = \kappa_i, \forall i = 1, ..., n$.
□

It is interesting to notice that the $Z_i$'s are no longer linear cost functions which is induced by the constraint of lot-streaming. They can be decomposed into two parts: a linear component made up of the earliness, tardiness and completion time of the job, and a non-linear component which depends on the number of sublots on the one hand and on the starting times of the sublots on the other hand. Intuitively, the latter component is minimised by increasing the number of sublots (ideally equal to $q_i$) and by reducing the gap between $t_{i,\pi_i(1),k}$ and $t_{i,\pi_i(2),k}$ (ideally there is no waiting time for the sublots). Therefore, solving such a problem involves solving three subproblems: the sequencing problem, the optimal timing problem and the sublot dimensioning problem. The first two problems are classics of the JiT literature whilst the last one is related to the lot-streaming constraint.

Let us consider the single-operation jobs problems. We show that solving the problem with divisible jobs is equivalent to solving the problem with indivisible jobs. We start with an instrumental lemma.

**Lemma 3.4:** *For the single-operation job problems with lot-streaming constraint* *there is no idle time between two sublots of the same job in any strict Pareto optimal schedule for the $Z_i$'s.*

**Proof:** As stated in corollary 3.2, the total cost $Z_i$ is given by
$Z_i = \beta_i T_i + \kappa_i q_i E_i + \delta\lambda_i + \kappa_i q_i C_{i,1} - \frac{q_i\kappa_i}{\delta}\left(\sum_{k=1}^{\delta} t_{i,1,k} + \frac{p_{i,1}}{2}\right)$.
Assuming that schedule $\sigma$ is a strict Pareto optimal schedule and is such that $\exists \ell > 1$ with $t_{i,1,\ell} > t_{i,1,\ell-1} + \frac{p_{i,1}}{\delta}$. Schedule $\sigma'$ can be built from $\sigma$ by right timeshifting the $(\ell-1)$th sublot of job $J_i$ until $t_{i,1,\ell} = t_{i,1,\ell-1} + \frac{p_{i,1}}{\delta}$. Consequently, $t_{i,1,\ell-1}(\sigma') > t_{i,1,\ell-1}(\sigma)$ and $Z_i(\sigma') < Z_i(\sigma)$ which implies a contradiction with the fact that $\sigma$ is a strict Pareto optimum.
□

**Theorem 3.5:** *The single machine and parallel machines JiT scheduling problems with divisible jobs reduce to the problems with indivisible jobs.*

**Proof:** From lemma 3.4 we can deduce that $t_{i,1,k} = C_{i,1} - (\delta - k + 1)\frac{p_{i,1}}{\delta}$ since there is no idle time between two consecutive sublots of the same job. Putting this result together with the total job cost established in corollary 3.3 the total job scheduling cost $Z_i$ can be rewritten as
$Z_i = \beta_i T_i + \kappa_i q_i E_i + \delta\lambda_i + \frac{q_i\kappa_i}{2} p_{i,1}$.
Each criterion $Z_i$ is therefore minimal when $\delta = 1$ whatever the corresponding schedule. Thus,minimising this $Z_i$ is equivalent to minimise the $Z_i$ stated in corollary 3.2. □

We now review the literature on JiT scheduling problems with the lot-streaming constraint. Quite a few such problems have been considered in the literature and most of the problems involving the lot-streaming constraint are devoted to the minimisation of the makespan for various shop configurations. We refer to Chang and Chiu (2005) for a recent survey on this topic.
Jin *et al.* (1999) tackle a jobshop problem with equal-size sublots and setup times. The aim is to minimize a quadratic penalty function $F$ of the promptness and tardiness of jobs, defined as $F = \sum_{i=1}^{n} \beta_i T_i^2 + \kappa_i P_i^2$, where $P_i$ is the promptness of job $J_i$ and is given by $P_i = \max(0, s_i - t_{i,\pi_i(1),1})$ with $s_i$ the target starting time for $J_i$. This last measure enables to reduce the storage of raw materials. Taking into account the promptness and tardiness aimed at reducing work-in-process storage costs and customer dissatisfaction. However, again, situations exist for which reducing the promptness is not equivalent to reducing neither the work-in-process storage costs nor the final product storage cost. This is mainly due to the fact that if a job starts after its target starting time, we have equivalent schedules with regards to the promptness whilst the work-in-process may differ. Exact and heuristic algorithms based on lagrangean relaxation and dynamic programming are proposed by Jin et al.
Yoon and Ventura (2002b) consider the early/tardy flowshop scheduling problem with lot-streaming, problem which is referred to as $F|d_i, lot-streaming|F_\ell(\bar{E}^\kappa, \bar{T}^\beta)$. They first focus on solving the optimal timing problem for equal-size sublots for various additionnal constraints like buffer capacities, no-wait constraint and blocking constraint. They also present a linear program for solving the optimal timing problem in case of the flowshop problem with consistent sublots and infinite capacity buffers. Yoon and Ventura (2002b) next propose simple neightbourhood heuristics for the general equal-size sublot flowshop problem, each fixed sequence being evaluated by solving the associated linear program. Yoon and Ventura (2002a) present a genetic algorithm for the sequencing problem. Later on, Tseng and Liao (2008) propose for this problem a particule swarm optimization algorithm which is experimentally shown to be outperforming this genetic

algorithm.
Despite the fact that the tackled problem includes the lot-streaming constraint and is very close from the minimization of the sum of the $Z_i$'s, it does not fully answer the requirements of a JiT production system since it does not take into account wastes induced by the storage of work-in-process. To illustrate this point, we can have a second look at the second example of section 3.1 and consider the problem with two equal-size sublots: the same conclusions apply.

## 4. New Just-in-Time scheduling models

In this section we investigate new possible JiT scheduling models which aim at being more realistic and which are based on the job cost functions defined in section 2.3. These models illustrate how to optimise simultaneously criteria $Z_i$ and how the decision maker's preferences can be classicaly tackled. Multicriteria optimisation and decision aid fields provide insights on how to model these preferences and on how they can be taken into account in the optimisation phase. In the remainder we apply these results to derive possible JiT scheduling models.

One must remember that the general multicriteria problem consists in minimizing the $n$ criteria $Z_i$ according to the decision maker's preferences. Basically, these preferences can be expressed more or less accurately by means, for each criterion $Z_i$, of a weight $\alpha_i$, a bound $\epsilon_i$ on the maximum allowed value or a goal $\mu_i$ to reach. In multicriteria optimisation literature there are numerous distinct methods to combine these preferences in order to calculate a single strict Pareto optimum and we only focus here on two basic ones, namely the convex combination of criteria and the parametric approach.
The *convex combination method*, referred to as $F_\ell(Z_1, ..., Z_n)$, consists in minimising a convex combination of the $Z_i$'s, *i.e.* $F_\ell(Z_1, ..., Z_n) = \sum_{i=1}^{n} \alpha_i Z_i$ with $\alpha_i \in [0; 1]$ and $\sum_{i=1}^{n} \alpha_i = 1$. In fact, this method is the one which is used for most of the JiT scheduling problems tackled in the literature, and the convex combination is a measure of the total JiT cost which have a sense in practice.
The *parametric approach*, referred to as $P(Z_1, Z_2, ..., Z_n)$, consists in minimising a strictly increasing function $g$ of the criteria subject to bound constraints of the form $Z_i \le \epsilon_i$, $\forall i = 1, ..., n$. It is convenient here to consider the following function $g = \sum_{i=1}^{n} \alpha_i Z_i$ with $\alpha_i$ weights defined as in the convex combination method. Therefore, this application of the parametric approach yields a more general model than the one based on a single convex combination of criteria. Besides, it keeps a sense in practice since we minimise the total JiT cost subject to bounds on individual job costs.

An important result relies on the capability to calculate strict Pareto optima, for the $Z_i$'s, for each of these two approaches. It is well-known (T'kindt and Billaut 2006) that minimising a convex combination of criteria for non-convex problem only enables to provide a decision maker with a subset of strict Pareto optima, namely the *supported strict Pareto optima*. In other words, if we minimise all possible convex combinations by trying all possible weights $\alpha_i$, we only obtain a subset of all the strict Pareto optima and some potentially interesting solutions for the decision maker may be missed. This does not hold anymore for the parametric approach since by also changing the bounds $\epsilon_i$ we can calculate both the *supported* and *non supported* strict Pareto optima.

To complete this section, we mention that a bound constraint $Z_i \le \epsilon_i$ means that we restrict the starting and the completion times of the last sublot of the last operation of job $J_i$, *if there is no work-in-process storage cost.* For instance, consider the single machine problem with indivisible jobs tackled in Estève *et al.* (2006) and referred to as $1|d_i|P(Z_1, ...Z_n)$ or equivalently $1|d_i, Z_i \leq \epsilon_i| \sum_i \kappa_i E_i + \beta_i T_i$.

**Lemma 4.1:** *The $1|d_i, Z_i \leq \epsilon_i| \sum_i \kappa_i E_i + \beta_i T_i$ problem, with indivisible jobs, is equivalent to the $1|r_i, d_i, \tilde{d}_i| \sum_i \kappa_i E_i + \beta_i T_i$ problem with $r_i = \max(0; d_i - p_i - \lfloor \epsilon_i/\kappa_i \rfloor)$ and $\tilde{d}_i = d_i + \lfloor \epsilon_i/\beta_i \rfloor$, $\forall i = 1, ..., n$. We call $r_i$ the cost related release date and $\tilde{d}_i$ the cost related deadline. We necessarily have $C_i \geq r_i$ and $C_i \leq \tilde{d}_i$.*

In Table 2 we provide some new JiT scheduling problems based on the application of the parametric approach on the $Z_i$'s. The first column refers to the shop environment, with specification as to whether the lot-streaming constraint is enabled or not. The second column provides the total cost function defined as $\sum_{i=1}^{n} Z_i$ and the third column provides information about the cost related release times and deadlines deduced from the constraints $Z_i \leq \epsilon_i, \forall i = 1, ..., n$. Notice that whenever possible we indicate by $r_i$ and $\tilde{d}_i$ the exact values of the cost related data, whilst sometimes we can only establish a lower bound on the release date, denoted by $r_i^L$. This last column only provide available information about the scheduling problem. The cost related data for the single machine and parallel machines environments are given by Estève *et al.* (2006) for the case of indivisible jobs. These data do not change for the divisible jobs case due to lemma 3.4. Concerning the 2-machine shop environments it seems harder to establish the exact formulation of the cost related release dates and deadlines. However we can deduce a lower bound on the cost related release dates.

**Lemma 4.2:** *For the 2-machine environment with indivisible jobs, the constraint $Z_i \leq \epsilon_i$, $\forall i = 1, ..., n$, induces a lower bound $r_i^L$ on the cost related release date defined by*

$$r_i^L = d_i - p_{i,\pi_i(1)} - p_{i,\pi_i(2)} - \lceil \tfrac{1}{\kappa_i}(\epsilon_i - \gamma_i p_{i,\pi_i(2)}) \rceil - \max(0; \lceil \tfrac{\kappa_i - \beta_i}{\gamma_i} \rceil).$$

**Proof:** We assume, for any given $i$, that job $J_i$ completes early. In this case, according to corollary 3.2 the job cost is given by

$$Z_i = \kappa_i(d_i - C_{i,\pi_i(2)}) + \gamma_i(C_{i,\pi_i(2)} - C_{i,\pi_i(1)})$$

and the constraint $Z_i \leq \epsilon_i$ is equivalent to
$C_{i,\pi_i(2)}(\gamma_i - \kappa_i) - \gamma_i t_{i,\pi_i(1),1} - \gamma_i p_{i,\pi_i(1)} \leq \epsilon_i - \kappa_i d_i$
but as $C_{i,\pi_i(2)} \geq t_{i,\pi_i(1),1} + p_{i,\pi_i(1)} + t_{i,\pi_i(2)}$ the previous inequality implies
$-\kappa_i t_{i,\pi_i(1),1} \leq \epsilon_i - \kappa_i d_i + \kappa_i p_{i,\pi_i(1)} + (\kappa_i - \gamma_i) p_{i,\pi_i(2)}$
$\Rightarrow t_{i,\pi_i(1),1} \geq r_i^L = d_i - p_{i,\pi_i(1)} - p_{i,\pi_i(2)} - \frac{1}{\kappa_i}(\epsilon_i - \gamma_i p_{i,\pi_i(2)})$.
To complete, we just have to notice that this lower bound on the starting time remains valid even if job $J_i$ completes tardy and $\beta_i \geq \kappa_i$. If this last inequality does not hold, then the release time can be decreased by $\frac{\kappa_i - \beta_i}{\gamma_i}$. Thus we obtain the following lower bound on the release date

$$r_i^L = d_i - p_{i,\pi_i(1)} - p_{i,\pi_i(2)} - \lceil \tfrac{1}{\kappa_i}(\epsilon_i - \gamma_i p_{i,\pi_i(2)}) \rceil - \max(0; \lceil \tfrac{\kappa_i - \beta_i}{\gamma_i} \rceil).$$

□

It seems difficult to establish such a lower bound for the 2-machine environments with divisible jobs due to the lack of direct link between the cases early and tardy in the above proof.

Insert Table 2 about here

Table 2 proposes some new JiT scheduling problems to tackle for simple shop environments. For instance, solving the JiT parallel machines problem using the proposed parametric approach is equivalent to solving the $P|r_i, d_i, \tilde{d}_i|\bar{E}^\kappa + \bar{T}^\beta$ problem with the above cost related release dates and deadlines. This problem, when solved to optimality for given bounds $\epsilon_i$, enables to calculate a strict Pareto optima for the $Z_i$'s and therefore do not lead to the calculation of non desirable solutions for a decision maker. From Table 2 it appears that the multiple-operation environments already seem to harder the harder to tackle due to the difficulty to establish the equation of the cost related data.

## 5. Conclusions and future research directions

In this paper we have investigated JiT scheduling problems in the light of the existing JiT production literature, establishing a realistic formulation of the cost of scheduling jobs just-in-time. A review of the existing literature has shown that the basic early/tardy models are almost all dominated by the proposed approach in the sense that we can only benefit from using the new models instead of the classic ones considered in the literature. We have also considered the case when jobs can be divided into sublots, establishing that the lot-streaming constraint may help to reduce the storage costs and therefore answering the JiT production philosophy. **Another interest of the proposed models, is that they can be quite easily generalized to the case when the sublots are not necessarily of equal size, where there are setup times between sublots and where the transfer of a sublot between two machines requires a non negligible time.**

This paper sets out a lot of new research directions notably in the case of divisible jobs. Many open questions should be dealt with in the near future: How the existing optimal timing algorithms can be applied when lot-streaming is enabled? How can the lot-streaming problem be solved efficiently when the total JiT cost is minimised? Works dealing with this last topic exist but, to the best of our knowledge, they almost exclusively refer to cases in which the makespan is minimised. One should notice that the lot-streaming problem can be solved in polynomial time for several lot-streaming configurations, as for instance, the cases of equal-size sublots or consistent sublots (see Dauzère-Pérès and Lasserre (1997), Yoon and Ventura (2002b)). Moreover, the JiT scheduling problem with lot-streaming could be heuristically solved through an iterative scheme similar to the one proposed by Dauzère-Pérès and Lasserre (1997) but with three steps at each iteration: first, solve the lot-streaming problem with a fixed schedule, second, solve the sequencing problem and, third, solve the optimal timing problem.

**Acknowledgements.** The author would like to thank the anonymous referees for their comments that helped to improve the quality of this paper.

## References

Baglin, G., Bruel, O., Garreau, A. and Greif, M., 2001. *Management Industriel et Logistique*. Economica.

Baker, K.R. and Scudder, G.D., 1990. Sequencing with earliness and tardiness penalties: a review. *Operations Research*, 38 (1), 22–36.

Baker, K. and Jia, D., 1993. A comparative study of lot streaming procedures. *OMEGA International Journal of Management Science*, 21 (5), 561–566.

Bulbul, K., Kaminsky, P. and Yano, C., 2004. Flow Shop Scheduling with Earliness, Tardiness and Intermediate Inventory Holding Costs. *Naval Research Logistics*, 51, 407–445.

Chang, J. and Chiu, H., 2005. A comprehensive review of lot-streaming. *International Journal of Production Research*, 43 (8), 1515–1536.

Chrétienne, P. and Sourd, F., 2003. PERT scheduling with convex cost functions. *Theoretical Computer Science*, 292, 145–164.

Dauzère-Pérès, S. and Lasserre, J.B., 1997. Lot Streaming in job-shop scheduling. *Operations Research*, 45 (4), 584–595.

Della Croce, F. and Trubian, M., 2005. Optimal idle time insertion in early-tardy parallel machines scheduling with precedence constraints. *Production Planning and Control*, 13, 133–142.

Estève, B., Aubijoux, C., Chartier, A. and T'kindt, V., 2006. A Recovering Beam Search algorithm for the single machine Just-in-Time scheduling problem. *European Journal of Operational Research*, 172 (3), 798–813.

Garey, M.R., Tarjan, R.E. and Wilfong, G.T., 1988. One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, 13 (2), 330–348.

Gordon, V., Proth, J.M. and Chu, C., 2002a. Due date assignment and scheduling: SLK, TWK and other due date assignement models. *Production Planning and Control*, 13 (2), 157–177.

Gordon, V., Proth, J.M. and Chu, C., 2002b. A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, 139 (1), 1–25.

Gordon, V., Proth, J.M. and Strusevich, V., 2004. Scheduling with Due-Date Assignment. Leung (2004) Chapman & Hall/CRC Computer and Information Science serie, Volume 1, chap. 21.

Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 287–326.

Hall, N.G. and Posner, M.E., 1991. Earliness-tardiness scheduling problems, I: Weighted deviation of completion times about a common due date. *Operations Research*, 39, 836–846.

Hendel, Y. and Sourd, F., 2007. An improved earliness-tardiness timing algorithm. *Computers and Operations Research*, 34, 2931–2938.

Jin, B., Luh, P. and Thakur, L., 1999. An effective optimization-based algorithm for job shop scheduling with fixed-size transfert lots. *Journal of Manufacturing Systems*, 18 (4), 284–300.

Kaminsky, P. and Hochbaum, D., 2004. Due-Date Quotation Models and Algorithms. Leung (2004) Chapman & Hall/CRC Computer and Information Science serie, Volume 1, chap. 20.

Kannan, V. and Tan, K., 2005. Just in Time, total quality management and supply chain management: understanding their linkages and impact on business performance. *Omega*, 33, 153–162.

Leung, J.T., ed. , 2004. *Handbook of Scheduling: Algorithms, Models and Performance Analysis*. Chapman & Hall/CRC Computer and Information Science serie, Volume 1.

Nollet, J., Kélada, J. and Diorio, M., 1994. *La gestion des opérations et de la production*. Gatan Morin.

Pan, Y. and Shi, L., 2005. Dual constrained single machine sequencing to minimize total weighted completion time. *IEEE Transactions on Automation, Science and Engineering*, 2 (4), 344–357.

Pinedo, M. and Chao, X., 1999. *Operations Scheduling with applications in manufacturing and services*. Boston: Mc Graw Hill.

Potts, C. and van Wassenhove, L., 1994. Integrating Scheduling with Batching and Lot-Sizing: a Review of Algorithms and Complexity. *Journal of the Operational Research Society*, 43 (5), 395–406.

Ramasesh, R., Fu, H., Fong, D. and Hayya, J., 2000. Lot streaming in multistage production systems. *International Journal of Production Economics*, 66, 199–211.

Sarin, S. and Jaiprakash, P., 2007. *Flow Shop Lot Streaming*. Springer.

Schonberger, J., 1982. *Japanese manufacturing techniques: Nine hidden lessons in simplicity*. The Free Press (New York).

Shi, L. and Pan, Y., 2003. Minimizing Job Shop Inventory With On-Time Delivery Guarantees. *Journal of Systems Science and Systems Engineering*, 12 (4), 449–469.

Sourd, F., 2005. Optimal timing of a sequence of tasks with general completion costs. *European Journal of Operational Research*, 165, 82–96.

Szwarc, W. and Mukhopadhyay, S.K., 1995. Optimal Timing Schedules in Earliness-Tardiness Single Machine Sequencing. *Naval Research Logistics*, 42, 1109–1114.

T'kindt, V. and Billaut, J.C., 2006. *Multicriteria Scheduling. Theory, Models and Algorithms*. Heidelberg: Springer.

Triesch, D. and Baker, K., 1994. Basic Techniques for lot streaming. *Operations Research*, 41, 1065–1076.

Tseng, C.T. and Liao, C.J., 2008. A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. *European Journal of Operational Research*, 191 (2), 360–373.

Yoon, S. and Ventura, J., 2002a. An application of genetic algorithms to lot-streaming flow shop scheduling. *IIE Transactions*, 34, 779–787.

Yoon, S. and Ventura, J., 2002b. Minimizing the mean weighted absolute deviation from due dates in lot-streaming flowshop scheduling. *Computers & Operations Research*, 29, 1301–1315.

Table 1. Summary of the scheduling data

| | |
|---|---|
| $n$ | : number of jobs, |
| $m$ | : number of machines, |
| $\delta$ | : number of sublots per operation, |
| $q_i$ | : number of indivisible elements of job $J_i$, |
| $\lambda_i$ | : cost for dividing job $J_i$ in elements, |
| $r_i$ | : release date of job $J_i$, |
| $d_i$ | : due date of job $J_i$, |
| $\beta_i$ | : unit cost for completing job $J_i$ tardy (penalty costs), |
| $p_{i,j}$ | : processing time of the $q_i$ elements of $J_i$ on machine $M_j$, |
| $\gamma_i^{\pi_i(j)}$ | : unit storage cost of work-in-processes of job $J_i$ between machines $M_{\pi_i(j)}$ and $M_{\pi_i(j+1)}$, |
| $\kappa_i$ | : unit storage cost of finish products. |

Table 2. New multicriteria JiT scheduling problems

| Shop environment | | Total cost $\sum_i Z_i$ | Cost related data |
|---|---|---|---|
| Single machine and parallel machines | Indiv | $\bar{T}^{\beta} + \bar{E}^{\kappa}$ | $r_i = \max(0; d_i - p_i - \lfloor \epsilon_i/\kappa_i \rfloor)$ and $\tilde{d}_i = d_i + \lfloor \epsilon_i/\beta_i \rfloor$ |
| | Div | $\bar{T}^{\beta} + \bar{E}^{\kappa}$ | $r_i = \max(0; d_i - p_i - \lfloor \epsilon_i/\kappa_i \rfloor)$ and $\tilde{d}_i = d_i + \lfloor \epsilon_i/\beta_i \rfloor$ |
| 2-machine shop | Indiv | $\bar{T}^{\beta} + \bar{E}^{\kappa} + \bar{C}^{\gamma} - \sum_i \gamma_i C_{i,\pi_i(1)}$ | $r_i^L = d_i - p_{i,\pi_i(1)} - p_{i,\pi_i(2)} - \lceil \frac{1}{\kappa_i}(\epsilon_i - \gamma_i p_{i,\pi_i(2)}) \rceil - \max(0; \lceil \frac{\kappa_i - \beta_i}{\gamma_i} \rceil)$ |
| | Div | $\bar{T}^{\beta} + \bar{E}^{\kappa} + \bar{C}^{\gamma q} - \sum_i \frac{q_i}{\delta} ( \sum_{k=1}^{\delta} (\gamma_i t_{i,\pi_i(1),k} + (\kappa_i - \gamma_i) t_{i,\pi_i(2),k}) + \frac{\gamma_i p_{i,\pi_i(1)} + (\kappa_i - \gamma_i) p_{i,\pi_i(2)}}{2})$ | |

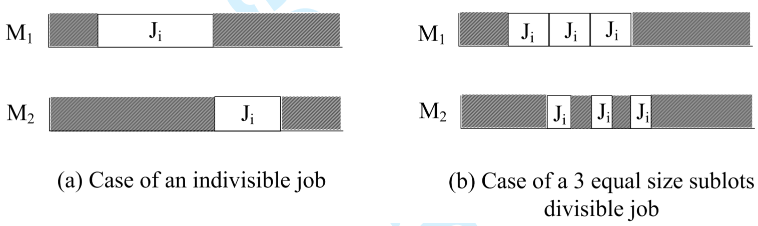



Figure 1. An example of a divisible job and of an indivisible job

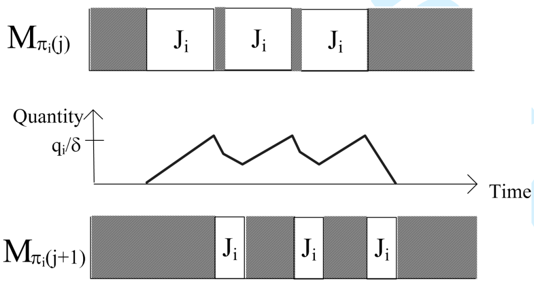



Figure 2. An example of a work-in-process storage area evolution

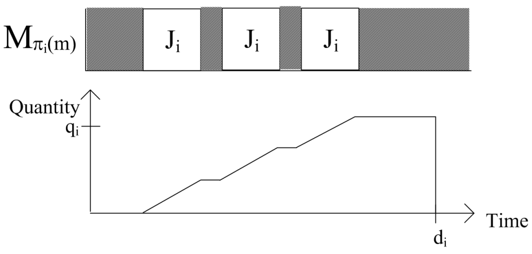



Figure 3. An example of a final product storage area evolution

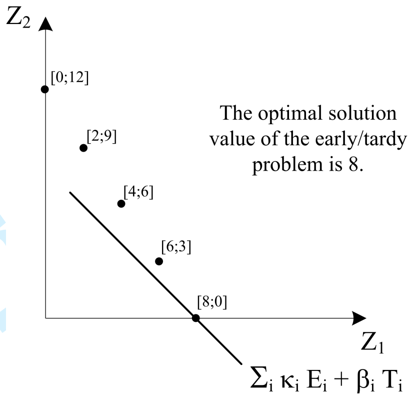



Figure 4. A single machine sample early/tardy scheduling problem

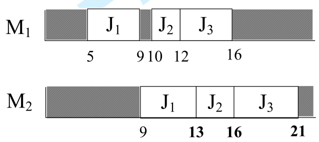



Figure 5. A 2-machine flowshop sample early/tardy scheduling problem

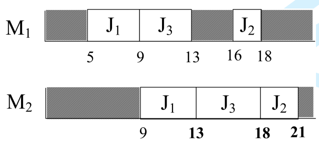
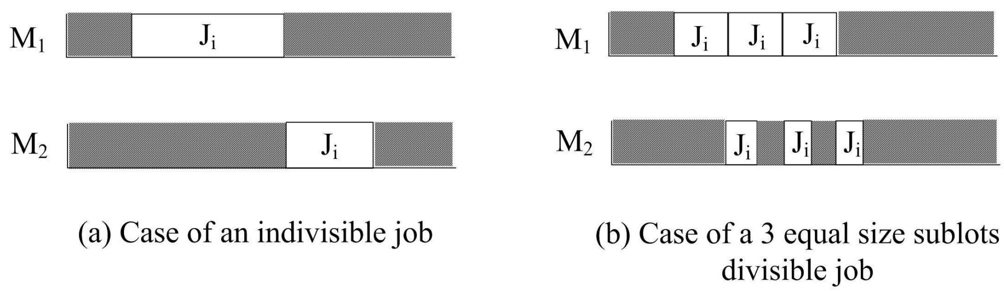
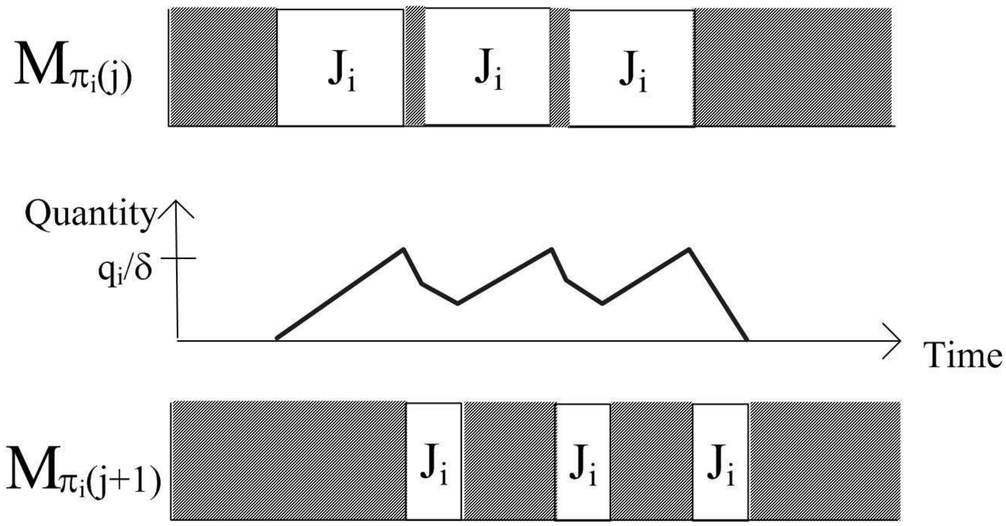


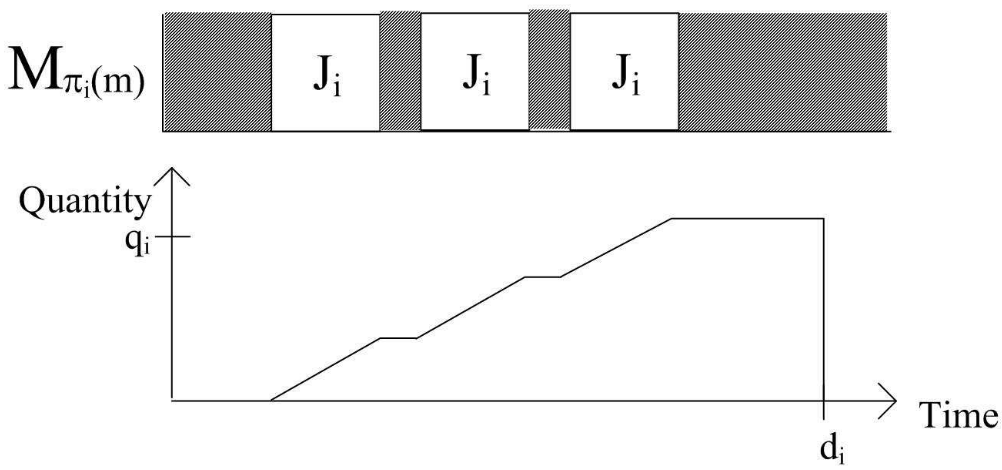

Figure 6. A 2-machine flowshop sample early/tardy scheduling problem



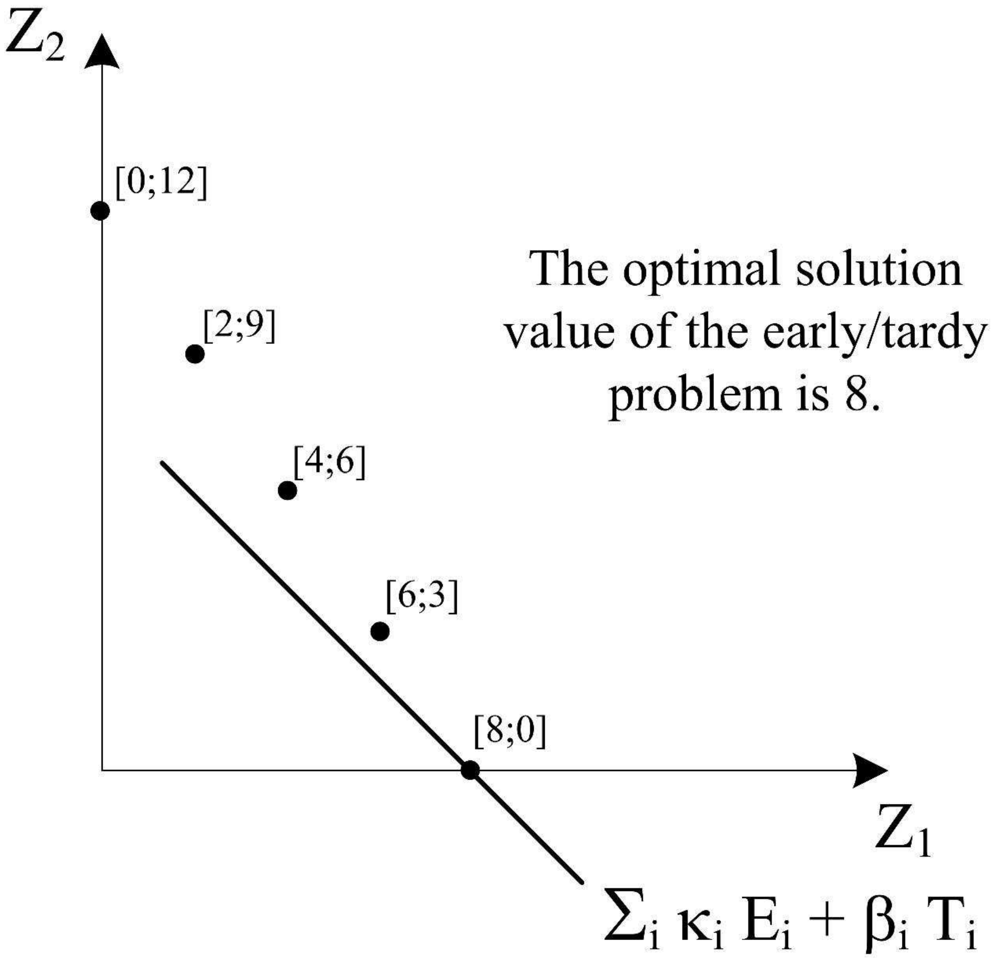

An example of a divisible job and of an indivisible job
92x25mm (600 x 600 DPI)



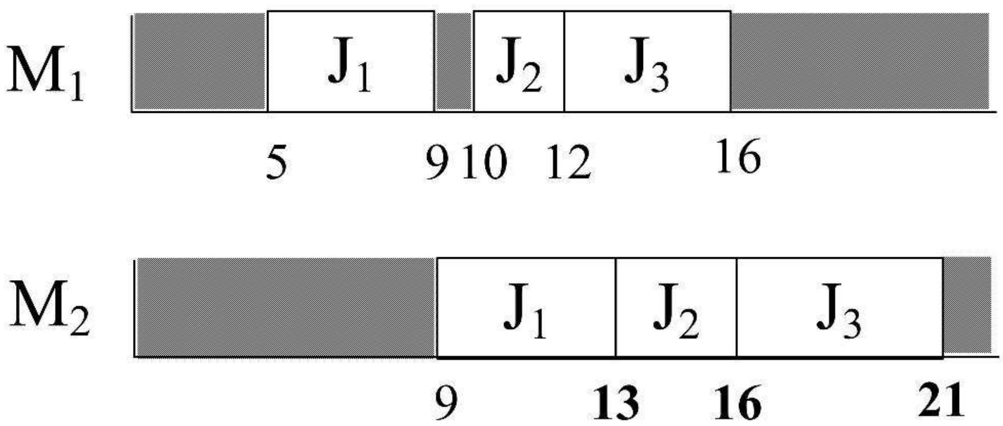

An example of a work-in-process storage area evolution
55x28mm (600 x 600 DPI)



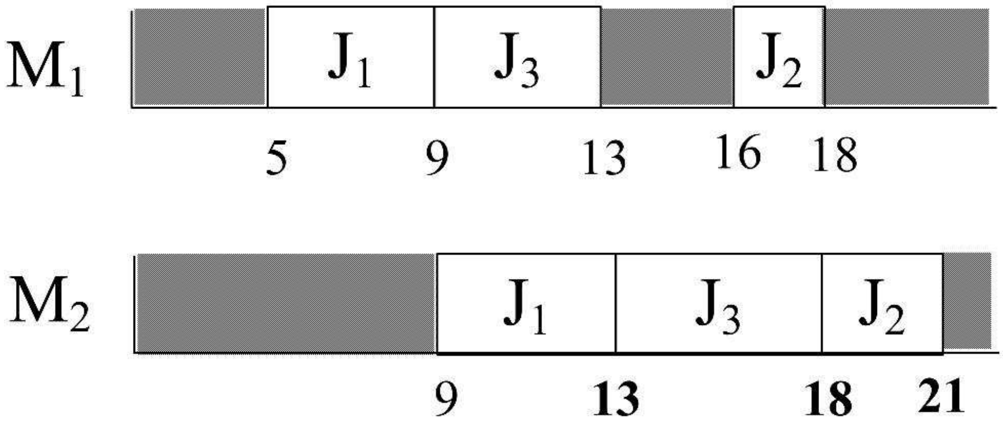

An example of a final product storage area evolution
55x25mm (600 x 600 DPI)




A single machine sample early/tardy scheduling problem
53x53mm (600 x 600 DPI)




A 2-machine flowshop sample early/tardy scheduling problem
41x17mm (600 x 600 DPI)




A 2-machine flowshop sample early/tardy scheduling problem
41x17mm (600 x 600 DPI)